Vamos a crear un nuevo proyecto que se llamará HolaUsusario. Como requisitos del mismo tenemos: queremos que se ejecute en modo apaisado y que no le afecte el cambio de orientación del dispositivo. La versión mínima soportada será la 15 y el target para la 26. La aplicación constará de dos pantallas, en la primera el usuario introducirá su nombre y cuando el usuario pulse el botón *Saludo*, se abrirá otra ventana donde saludaremos de forma personalizada al usuario.

PASO1. Crearemos el proyecto siguiendo los pasos ya realizados en los ejercicios anteriores

PASO2. Pasemos a definir el layout de la activity principal

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <TextView android:text="Usuario:"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"

        android:ems="10"
        android:id="@+id/editText"
        android:background="#ff813f" />

    <Button
        android:layout_width="149dp"
        android:layout_height="wrap_content"
        android:text="Saludo"
        android:id="@+id/button"
        android:layout_gravity="center_horizontal" />
</LinearLayout>
```

El resultado visual de la primera ventana de nuestra aplicación sería algo parecido al siguiente:

PASO3. Pasemos a definir la interfaz gráfica de la segunda ventana. Para ello creamos un nuevo fichero llamado *activity2.xml*, y en él añadimos únicamente una etiqueta *TextView* donde mostraremos el mensaje personalizado al usuario.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"

        android:id="@+id/textView"
        android:layout_gravity="center_horizontal" />

</LinearLayout>
```

PASO4. Crear una activity para la pantalla secundaria. Crearemos una clase nueva derivada de AppCompactActivity. En ella implementaremos el método *onCreate*, al igual que se hizo para la anterior.

```
public class Activity2 extends AppCompatActivity{
    TextView texto;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity2);
        texto=(TextView)findViewById(R.id.textView);
       /* Bundle b= this.getIntent().getExtras();
        texto.setText( b.getString("SALIDA"));*/
        texto.setText(getIntent().getStringExtra("SALIDA"));
    }
```

Posteriormente pasaremos a explicar el código correspondiente al paso de información entre las activitys.

Ahora es necesario especificar la nueva activity en el fichero de manifiesto.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Resuelto2Activitys"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="Resuelto2Activitys" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity android:name=".Activity2"/>
</application>
```

PASO5. Vamos a implementar la lógica de funcionamiento de nuestra aplicación. En primer lugardebemos controlar el acceso al cuadro de texto y al botón de nuestra pantalla principal. Para ello necesitamos crear dos variables en la activity y utilizaremos el método *findViewById()* , pasándole el ID de cada control y que están definidos en la clase *R.*

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
    EditText nombre;
    Button saludo;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        nombre=(EditText)findViewById(R.id.editText);
        saludo=(Button)findViewById(R.id.button);

        saludo.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        if(v.getId()==R.id.button){
            Intent intento=new Intent(this,Activity2.class);
           /* Bundle bundle=new Bundle();
            bundle.putString("SALIDA", nombre.getText().toString());
            intento.putExtras(bundle);*/
            intento.putExtra("SALIDA","Hola "+ nombre.getText());
            startActivity(intento);

        }
    }
}
```

Para comunicar a la otra activity (pantalla secundaria) que se visualice, lo haremos a través de un intent. Para ello le pasamos al constructor una referencia de la propia activity llamadora (*MainActivity*) y la clase de la activity invocada (*MiSaludo.class).*

Intent comunicador=new Intent(MainActivity.this,Activity2.class);

Intent comunicador=new Intent(this,Activty2.class);

Existen distintas variantes del constructor de la clase Intent, cada una de ellas dirigida a unas determinadas acciones, en nuestro caso vamos a definir un intent para llamar a una activity dentro de la misma aplicación.

Como tenemos que pasarle información a la activity (el mensaje que el usuario ha tecleado) podemos hacerlo utilizando un objeto *Bundle*, que puede contener una lista de pares clave-valor con toda la información a pasar entre actividades. En este ejemplo hemos añadido un único dato de tipo *String* mediante el método *putString(clave,valor*). Tras ello añadimos la información al *intent* mediante el método *putExtras(bundle)*. Cuando deseamos desempaquetar el dato en la próxima clase, tendremos que utilizar la misma clave.

```
//Creamos el objeto bundle y le añadimos el par valor
Bundle datosAComunicar=new Bundle();
datosAComunicar.putString("SALUDO","Hola
"+editeTextNombre.getText().toString()+" "+"Vienvenido a la
aplicación");
//Añadimos al intent el bundle
comunicador.putExtras(datosAComunicar);
```

O podemos hacerlos directamente añadiendo el dato al Intente, como en el código de la aplicación:

```
Intent intento=new Intent(this,Activity2.class);
intento.putExtra("SALIDA","Hola "+ nombre.getText());
```

Ahora podemos pasar a activar la activity.

```
//Iniciamos la nueva actividad
startActivity(intento);
```

PASO6. Veamos ahora que hacer en la actividad secundaria

```
public class Activity2 extends AppCompatActivity{
    TextView texto;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity2);
        texto=(TextView)findViewById(R.id.textView);
       /* Bundle b= this.getIntent().getExtras();
        texto.setText( b.getString("SALIDA"));*/
        texto.setText(getIntent().getStringExtra("SALIDA"));
    }
```

Debemos recuperar la información pasada desde la activity principal y asignarla como texto de la etiqueta. En primer lugar definir la variable asociada al TextView. A continuación recuperar la información en un objeto Bundle si hemos usado este modo (está entre comentarios) o directamente que es el usado en esta solución

Bundle bundle = getIntent().getExtras();

A continuación se construye el texto de la etiqueta a través del método *setText(texto)* .

Con esto hemos concluido el desarrollo de nuestra aplicación, tan solo nos falta ejecutarla y ver cómo funciona.